



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/593,262	09/18/2006	Yves Gattegno	200800984-6	6443
22879	7590	11/26/2010	EXAMINER	
HEWLETT-PACKARD COMPANY Intellectual Property Administration 3404 E. Harmony Road Mail Stop 35 FORT COLLINS, CO 80528				PATEL, SHAMBHAVI K
ART UNIT		PAPER NUMBER		
2128			NOTIFICATION DATE	
11/26/2010			DELIVERY MODE	
ELECTRONIC				

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

JERRY.SHORMA@HP.COM
ipa.mail@hp.com
laura.m.clark@hp.com

Office Action Summary	Application No.	Applicant(s)
	10/593,262	GATTEGNO, YVES
	Examiner	Art Unit
	SHAMBHAVI PATEL	2128

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) Responsive to communication(s) filed on 09/01/10.
 2a) This action is FINAL. 2b) This action is non-final.
 3) Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) Claim(s) 1-46 is/are pending in the application.
 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
 5) Claim(s) _____ is/are allowed.
 6) Claim(s) 1-46 is/are rejected.
 7) Claim(s) _____ is/are objected to.
 8) Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) The specification is objected to by the Examiner.
 10) The drawing(s) filed on _____ is/are: a) accepted or b) objected to by the Examiner.
 Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
 Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
 11) The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
 a) All b) Some * c) None of:
 1. Certified copies of the priority documents have been received.
 2. Certified copies of the priority documents have been received in Application No. _____.
 3. Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- | | |
|--------------------------------------------------------------------------------------|-------------------------------------------------------------------|
| 1) <input type="checkbox"/> Notice of References Cited (PTO-892) | 4) <input type="checkbox"/> Interview Summary (PTO-413) |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948) | Paper No(s)/Mail Date. _____ . |
| 3) <input type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08) | 5) <input type="checkbox"/> Notice of Informal Patent Application |
| Paper No(s)/Mail Date _____. | 6) <input type="checkbox"/> Other: _____ . |

DETAILED ACTION

1. Claims 1-46 have been presented for examination.

Response to Arguments

2. Applicant's arguments regarding the prior art rejection have been fully considered, but are not persuasive.

- i. **Applicant submits**, on pages 14-15, that Rissmeyer discloses a fixed order of initially loading a network driver and then loading a disk driver, but does not disclose that this predetermined loading of components may be modified.

Examiner notes that Applicants appear to be arguing that the a predetermined order or sequence of loading and execution of components of an operating system changes or may be modified *after* the real hard disk representation is created. However, the claims only require that they be modified/changed *in order to create* the real hard disk representation. Accordingly, Rissmeyer teaches modifying the sequence for loading and executing of components of the operating system when booting from a bootable operating system (**see column 1 lines 46-59**) by loading a network driver before loading a disk driver.

- ii. **Applicant submits**, on page 15, that Zimmerman discloses that the address of a server to boot from is provided to a client 2 PC upon power-up of the client 2PC (from hibernation), but does not disclose that the location for loading and execution of components may be modified.

Examiner notes that Applicants appear to be arguing that Zimmerman does not disclosing modifying the *destination* location of the components. However, the claim only states "wherein...the location for loading and execution of components...may be modified." The claims are given their broadest reasonable interpretation, and accordingly, the "location" in the claim may be the *source location or destination location* of the components. Specifically, Zimmerman discloses that the address of the server that is to be booted from maybe be modified ([0058]). This equivalent to modifying the source location of the components.

- iii. **Applicant submits**, on pages 16-17, that Zimmerman makes no mention of server being unable to accept writing in real-time, and makes no mention of any response if the server is unable to accept

writing in real time, much less a response to alter the storage destination if the server is unable to accept writing in real time.

Examiner notes that as per **paragraph [0069]**, all write requests are queued (i.e. no real-time writing requests are accepted). Specifically, when a write request is made to the O/S disk, the request is not immediately sent to the disk -- instead, the write request is diverted to a cache in the client system's memory (**paragraph [0067]**). These requests are then, at a later time, sent to the O/S disk, and the data is then returned to the O/S (**paragraph [0067]**). Therefore, the server does not accept write requests as they are made (i.e. real-time) in order to prevent different clients from simultaneously writing to the same image and corrupting it (**paragraph [0069]**). That these requests are instead cached in a local memory is equivalent to the claimed response to alter the storage destination.

- iv. **Applicant submits**, on pages 17-18, that Zimmerman does *not* disclose an emulated hard disk carried out in *different* storage spaces, because the single server streams the master boot record.
Examiner notes that the broadest reasonable interpretation of claims 19-20 only require that the read operation be performed in different storage spaces -- not that the ultimate source of transmit of the read data be different. As disclosed in **paragraph [0019]**, the network server, which is where the disk is stored, may be implemented in a multi-server system. Therefore, when a read request to an I/O disk is made, the location of the requested data may be on any of the servers, thereby disclosing that the location of the read request is performed in different storage spaces. That the data is then streamed back from a single source does not preclude the aforementioned disclosure from teaching the claim limitation.
- v. **Applicant submits**, on page 18, that the invitation system in **paragraph [62]** cannot be reasonably characterized as the recited *order of priority*.
Examiner notes that Applicant has provided no justification for this conclusion. When a first client is selected as a Read Requestor, only that client is allowed to make a read request ([0062]), - thus, that client's read request is given priority over other client's read requests.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

The factual inquiries set forth in *Graham v. John Deere Co.*, 383 U.S. 1, 148 USPQ 459 (1966), that are applied for establishing a background for determining obviousness under 35 U.S.C. 103(a) are summarized as follows:

1. Determining the scope and contents of the prior art.
2. Ascertaining the differences between the prior art and the claims at issue.
3. Resolving the level of ordinary skill in the pertinent art.
4. Considering objective evidence present in the application indicating obviousness or nonobviousness.

This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).

3. Claims 1-46 are rejected under 35 U.S.C. 103(a) as being unpatentable over Zimmerman (US Pub. No. 2003/0200290) in view of Rissmeyer (US Patent No. 6,857,069).

Regarding claim 1:

Zimmerman discloses a method, performed by a suitably programmed computer, for software emulation of hard disks of a data processing platform at the level of the operating system with parameterizable management of requests for writing and reading data, the method comprising:

- a. creating a representation of a real hard disk wherein the location for loading and execution of components of the operating system of the data processing platform may be modified ([0019]: **drives hold O/S; [0058]: drives may comprise any nonvolatile storage devices**)

- b. loading on said data processing platform one or more peripheral drivers (**[0044]: storage driver and network filter driver**), wherein at least one of the peripheral drivers communicates with a data storage support containing the data of the representation of the real hard disk (**[0044]: storage driver and network filter driver used to download files from the virtual drive**).
- c. simulating the behavior of the real hard disk for the operating system wherein the method transforms programming contained on the real hard disk into an emulated hard disk (**[0026]: “virtual” storage device**) capable of controlling read and write operations on a client station and among two or more client stations (**[0026]: requests for data result in transparent emulation of operations of local disk at each of the clients; [0022]: one or more client devices**)

Zimmerman does not explicitly disclose modifying the sequence for loading and executing of components of the operating system of the data processing platform may be modified. **Rissmeyer teaches** modifying the sequence for loading and executing of components of the operating system of the data processing platform may be modified (**column 1 lines 46-59: loading a network driver before loading a disk driver in an OS booting over a network**). At the time of the invention, it would have been obvious to one of ordinary skill in the art to combine the teachings of Zimmerman and Rissmeyer because this allows the traditional controlling of devices prior to booting to occur without customized parts (**Rissmeyer: column 1 lines 11-44**).

Regarding claim 2:

Zimmerman discloses the method of claim 1 wherein the management of said data write requests that the operating system sends to the emulated hard disk is accomplished at the peripheral driver level (**[0044]: drivers**), the written data being stored according to the parameterization of the peripheral drivers, and wherein the parameterization accommodates the written data being stored in the memory accessible to the OS using the emulated hard disk (**[0069]: writes locally cached**).

Regarding claim 3:

Zimmerman discloses the method as claimed in claim 1 wherein the management of said data read requests that the operating system sends to the emulated hard disk is accomplished at the peripheral driver level

([0044]: **storage driver and network filter driver used to download files from the virtual drive**), the readings of previously written data being performed in the nonvolatile storage space accessible to the operating system using the emulated hard disk (**[0058]: nonvolatile**)

Regarding claim 4:

Zimmerman discloses the method as claimed in claim 1 wherein the emulation of the hard disk is accomplished by the agency of a single monolithic peripheral driver which communicates with the OS in the manner of a hard disk and which communicates with the support containing the data of said emulated hard disk in a manner specific to this support (**[0055]: single driver for some systems**).

Regarding claim 5:

Zimmerman discloses the method as claimed in claim 1, wherein the data of the emulated hard disk or disks are accessible to the client station via a data processing network and wherein the emulated hard disk comprises one or more emulated hard disks (**[0020]: network; [0022] and [0028]: multiple clients each with an emulated local disk**).

Regarding claim 6:

Zimmerman discloses the method as claimed in claim 1, wherein when an emulated hard disk is started up, the method further comprises a low level micro-software module to access data contained in the emulated hard disk wherein the operating system is started up at the client station (**[0020]; [0058] microinstruction code**).

Regarding claim 7:

Zimmerman discloses the method as claimed in claim 6, wherein the data processing network comprises a plurality of client stations, (**[0022]: multiple client devices**), the client stations using a bootup PROM (**[0024]: option PROM**), the method further comprising using the bootup PROM to control communications via the data processing network (**[0045]: communicate using PXE service**).

Regarding claim 8:

Zimmerman teaches the method as claimed in claim 7 wherein the low level micro-software module is loaded in the memory of the client station and executed using PROM ([0020]: **microinstruction code; [0026] downloading of code**).

Regarding claim 9:

Zimmerman teaches the method as claimed in claim 6, wherein the low level micro software module is loaded in memory of the client station and executed as a component of BIOS of the client station, said micro software module providing the same functions as the access serviced on real hard disks provided by the BIOS ([0020]: **microinstruction code; [0026] downloading of code; [0047]: BIOS**).

Regarding claim 10:

Zimmerman discloses a method as claimed in claim 6, wherein the low-level micro-software is loaded in memory of the client station from a third party data support supported as a startup peripheral by the client station ([0020]: **microinstruction code; [0026] downloading of code**).

Regarding claim 11:

Zimmerman discloses the method as claimed in claim 5, wherein at least one peripheral driver loaded and executed by the operating system of the client station provided the functions of access, via the data processing network, to the data contained in the emulated hard disks ([0044]: **storage driver and network filter driver used to download files from the virtual drive**).

Regarding claim 12:

Zimmerman discloses the method as claimed in claim 1 wherein if the data support does not provide for writing in real time, or does not accept the writing of data directly in the data of the hard disk, the data writing requests issued by the operating system to the emulated hard disks are processed in a way such that the written data are stored in a storage space different from the data support containing the data of the emulated hard disks ([0069]:

writes cached locally to prevent corruption of data) and wherein the emulated hard disk comprises one or more emulated hard disks (**[0022] and [0028]: multiple clients each with an emulated local disk**)

Regarding claim 13:

Zimmerman discloses the method as claimed in claim 12 wherein the data writing requests issued by the client station operating system to the emulated hard disks are processed in such a way that the written data are stored in the RAM of the client station (**[0069]: writes cached locally to prevent corruption of data**).

Regarding claim 14:

Zimmerman discloses the method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disks are processed in such a way that the written data are stored in the virtual memory of the client station (**[0069]: writes cached locally to prevent corruption of data**).

Regarding claim 15:

Zimmerman discloses the method as claimed in claim 12 wherein the data writing requests issued by the client station operating system to the emulated hard disks are processed in such a way that the written data are stored in a data file accessible to the operating system of the client station (**[0069]: writes cached locally to prevent corruption of data**).

Regarding claim 16:

Zimmerman discloses the method as disclosed in claim 1 wherein the data writing requests issued by the operating system are redirected to a single storage space, and wherein the storage space in which the written data are redirected may be changed during an operating session of the operating system of a client station (**[0021]: client module**).

Regarding claim 17:

Zimmerman discloses the method claimed in claim 12, wherein the storage space used for the storage may be volatile, or nonvolatile so as to permit the written data of an operating session of the operating system to persist from one client station to another (**[0022]: client includes local memory, ROM, RAM, local storage**).

Regarding claim 18:

Zimmerman discloses the method as claimed in claim 1 wherein the volatile character of the redirections of the written data is determined upon initialization of the operating session of the operating system of a client station (**[0021]: client module**).

Regarding claim 19:

Zimmerman discloses the method as claimed in claim 1, wherein the data reading requests issued by the operating system are performed in different storage spaces during an operating session of the OS of a client system (**[0021]: multi-server environment**).

Regarding claim 20:

Zimmerman discloses the method as claimed in claim 19, wherein the data reading requests issued by the OS to an emulated hard disk carried out in different storage spaces follow an order of priority (**[0062]**).

Regarding claim 21:

Zimmerman discloses the method as claimed in claim 5, wherein a server program is in charge at one of the client stations of the data processing network, on the one hand, of the communications via the data processing network with the client station accessing the emulated hard disks, and on the other, of accessing the data support containing the data of the emulated hard disks (**[0026]: transparent emulation of operations of local disk at each of the clients; “virtual” storage device”**)

Regarding claim 22:

Zimmerman discloses the method as claimed in claim 21, wherein if the hard disk emulation system is parameterized so that the data write requests received by the server program are intended for a specific emulated hard disk they are not redirected but stored directly in a support containing the data of the emulated hard disk itself and only client station can access said emulated hard disk at a given time (**[0069]: storage driver caches locally all writes so that the writes are never committed to the virtual drive, in order that different clients to no simultaneously write to the same virtual image and corrupt it.**)

Regarding claim 23:

Zimmerman discloses the method of claim 21, wherein in order to permit several client stations to access an emulated hard disk simultaneously, the server program is capable of redirecting specifically the data write requests issued by a client station to a given storage space and of redirecting the data write requests issued by another client station to another given storage space (**[0027]: simultaneous access by different clients of either same or different data).**

Regarding claim 24:

Zimmerman discloses the method as in claim 21 wherein in order to permit the startup form and/or simultaneous access to the same emulated hard disk of 100% identical copies of the same emulated hard disk, components of the operating system loaded and executed by the client stations or server programs are capable of modifying during or before their use by the operating system data contained in the emulated hard disk **[0012]: software upgrades only performed on a single disk image.**

Regarding claim 25:

Zimmerman discloses performing emulation for the OS of the client stations at the level of the class of virtual peripherals of the file system type (**[0026]: emulation**).

Regarding claim 26:

Zimmerman discloses performing the emulation for the OS at the level of the class of disk peripherals and not at the file system level (**[0026]: emulation**).

Regarding claim 27:

Zimmerman discloses the method as claimed in claim 1, wherein the data contained in the emulated hard disk are copied by a software tool executed at a client station from the real hard disk (**[0010]: download**).

Regarding claims 28 and 29:

Zimmerman discloses the method as claimed in claim 27 wherein the software tool creates an image file and directory that contains the data of the emulated hard disk (**[0010]: downloading of image files**).

Regarding claim 30:

Rissmeyer teaches modifying the loading sequence so that all components of the OS are loaded and usable at the moment when the OS needs to access the emulated hard disk by the peripheral drivers (**column 1 lines 46-59: loading a network driver before loading a disk driver in an OS booting over a network**).

Regarding claim 31:

Zimmerman discloses the method as claimed in claim 21 wherein in order to accelerate the simultaneous access by several client stations to the same emulated hard disk whose data are contained in a data support accessible to a server station, the data are sent by the server station to the client stations using the “broadcast” or “multicast” mechanisms (**[0010]: broadcast, multicast**).

Regarding claim 32:

Zimmerman teaches storing the streamed data by the client station in a local cache situated in the memory of said client stations (**[0032]: data stored in memory**).

Regarding claim 33:

Zimmerman teaches the method as claimed in claim 31 wherein a read request for data the in emulated hard disk issued by the operating system of a client station generates an explicit data reading request sent to the server station only if said data are not already present in the local cache (**[0069]: reads read from virtual drive unless a copy of the data has already been cached on the client**).

Regarding claim 34:

Zimmerman teaches the method as claimed in claim 33 wherein the data in the local cache are removed after being read by the client station so as to free up space in said local cache (**[0039]: cache emptied after data is no longer needed**).

Regarding claim 35:

Zimmerman teaches the method as claimed in claim 31 wherein the decision to send data by multicast/broadcast is made at the server module level which provides the functionalities necessary for the hard disk emulation at the client stations (**[0010]: server performs broadcast, multicast**).

Regarding claim 36:

Zimmerman discloses the method as claimed in claim 31, wherein the client stations may modify their subscription to receiving the data sent via broadcast/multicast by the server station (**[0038]: can choose to send data to only one client**).

Regarding claim 37:

Zimmerman discloses the method as claimed in claim 32, wherein the client stations may erase the data form the local cache after a certain parameterizable time (**[0039]: cache emptied after data no longer needed**).

Regarding claim 38:

Zimmerman discloses the method as claimed in claim 5, wherein data contained in a server module making the hard disks available to client stations may use any suitable network protocol ([0020]: **network**).

Rissmeyer teaches using the iSCSI protocol (**abstract**).

Regarding claim 39:

Zimmerman discloses the method as claimed in claim 5, wherein the low level software program executed by the client stations and permitting access to the data contained in the emulated hard disks may use any suitable network protocol ([0020]: **network**). **Rissmeyer teaches** using the iSCSI protocol (**abstract**).

Regarding claim 40:

Zimmerman discloses the method as claimed in claim 11, wherein at least one peripheral driver may use any suitable network protocol ([0020]: **network**). **Rissmeyer teaches** using the iSCSI protocol (**abstract**).

Regarding claim 41:

Zimmerman discloses the method as claimed in claim 21 wherein if the data storage support does not provide writing in real time, or does not accept the writing of data directly in the data of the hard disk, the server program providing the emulation of the hard disk at the client stations processes the data write requests issued by the operating system in such a way that the written data are stored in a storage space different from the data support containing the data of the emulated hard disks ([0069]: **writes cached locally to prevent corruption of data**).

Regarding claim 42:

Zimmerman discloses the method of claim 21 wherein the data write requests issued by the client station operating system to the emulated hard disks are processed in such a way that the written data are stored in the random access memory of the server station ([0069]: **writes cached locally to prevent corruption of data**).

Regarding claim 43:

Zimmerman discloses the method as claimed in claim 21, wherein the data writing requests issued by the client station operating system to the emulated hard disks are processed in such a way that the written data are stored in the virtual memory of the client station (**[0069]: writes cached locally to prevent corruption of data**).

Regarding claim 44:

Zimmerman discloses the method as claimed in claim 21 wherein the data writing requests issued by the client station operating system to the emulated hard disks are processed in such a way that the written data are stored in a data file accessible to the operating system of the client station (**[0069]: writes cached locally to prevent corruption of data**).

Regarding claim 45:

Zimmerman discloses the method claimed in claim 21, wherein the storage space used for the storage may be volatile, or maybe be nonvolatile so as to permit the written data of an operating session of the operating system to persist from one client station to another (**[0022]: client includes local memory, ROM, RAM, local storage**).

Regarding claim 46:

Zimmerman discloses the method as claimed in claim 16 wherein the volatile character of the redirections of the written data is determined upon initialization of the operating session of the operating system of a client station (**[0021]: client module**).

Conclusion

4. **Examiner's Remarks:** Examiner has cited particular columns and line numbers in the references applied to the claims above for the convenience of the applicant. Although the specified citations are representative of the teachings of the art and are applied to specific limitations within the individual claim, other passages and figures may apply as well. It is respectfully requested from the applicant in preparing responses, to fully consider the references in their entirety as potentially teaching all or part of the claimed invention, as well as the context of the passage as taught by the prior art or disclosed by the Examiner. In the case of amending the claimed invention,

Applicant is respectfully requested to indicate the portion(s) of the specification which dictate(s) the structure relied on for proper interpretation and also to verify and ascertain the metes and bounds of the claimed invention.

5. Applicant's amendment necessitated any new ground(s) of rejection presented in this Office action.

Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

6. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Shambhavi Patel whose telephone number is (571) 272-5877. The examiner can normally be reached on Monday-Friday, 8:00 am – 4:30 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Kamini Shah can be reached on (571) 272-2279. The fax phone number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

SKP

/Kamini S Shah/
Supervisory Patent Examiner, Art Unit 2128

/Hugh Jones/

Primary Examiner, Art Unit 2128